

EXHIBIT F

5.0 Reference

This section describes the actual OLE Interfaces exposed, the definitions of the data structures used when passing data around, and the definitions of each class used internally by the driver.

5.1 Interfaces

Other than exposing the standard interfaces IUnknown and IClassFactory, and the custom, *extended* XMCSPi interfaces, there is only one interface left – the IXMC_DrvStub interface. This interface is used to register the driver, used to perform *core* XMCSPi operations, with the driver stub. The diagram below displays all interfaces exposed by the XMC Driver Stub Component.

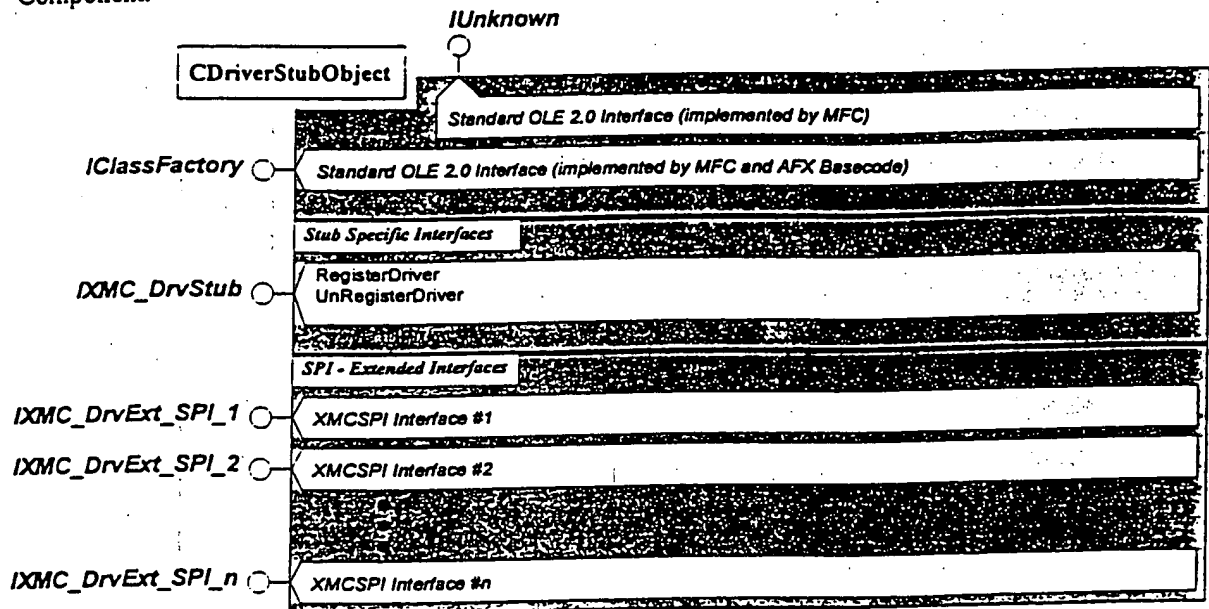


Figure 6 Interface-Map.

The section below describes the definition of the IXMC_DrvStub interface. For more information on the standard OLE interface used see the *OLE 2 Programmer's Reference*. And, for more information on the *extended* XMCSPi custom OLE interfaces see the *XMCSPi Reference*.

5.1.1 IXMC_DrvStub Interface

The following is the definition of the IXMC_DrvStub interface.

```

interface IXMC_DrvStub
{
    HRESULT RegisterDriver( LPCLSID lpDriverCLSID );
    HRESULT UnRegisterDriver( LPCLSID lpDriverCLSID );
};

```

5.2 Exported Functions

The following are the functions exported by the driver DLL.

```

XMC_DRIVERSTUB_MODULETYPE  DLLGetModuleType( void );
LPCLSID                    DLLGetCLSID( void );
BOOL                       DLLRegisterServer( void );
BOOL                       DLLUnRegisterServer( void );

```

5.3 Structures and Defines

This section defines all structures, enumerations, and defines used by the driver.

5.3.1 XMC_DRIVERSTUB_MODULETYPE Enumeration

The following enumeration is used to mark the module type of the driver stub.

```
enum XMC_DRIVERSTUB_MODULETYPE
{
    XMC_DRIVERSTUB_MT    = 0x1000
};
```

5.3.2 XMC_EXT_SPI Enumeration

The following enumeration is used as a function index for all *extended* XMCSPi functions implemented by the CSPIMgr object.

```
enum XMC_EXT_SPI
{
    XMC_ES_MOVEREL,
    :
};
```

5.3.3 XMC_CORE_SPI Enumeration

The following enumeration is used as a function index for all *core* XMCSPi functions accessed by the CSimpleDriver object.

```
enum XMC_EXT_SPI
{
    XMC_CS_MOVEABS,
    :
};
```

5.4 Classes

This section contains the definition of all classes used by the driver stub component in its implementation.

5.4.1 CDriverStubDisp Class

The CDriverStubDisp class acts as the dispatch, or control, object used to separate all OLE details from the code implementing the driver internals. This object coordinates all operations taking place in the driver by directing other C++ object to carry out the work of the operation. The following is the definition of the CDriverStubDisp class.

```
class CDriverStubDisp
{
public:
    //---- Constructors & Destructors ----

    CDriverStubDisp( void );
    ~CDriverStubDisp( void );

    //---- IXMC_DrvStub Methods ----

    DWORD RegisterDriver( LPCLSID lpDriverCLSID );
    DWORD UnRegisterDriver( LPCLSID lpDriverCLSID );

    //---- Actions ----

    DWORD FireExtSPI( XMC_EXT_SPI extspiID, ... )
```

```

private:
    //---- Private Data ----
    CSPIMgr      m_spiMgr;
};

```

5.4.2 CSPIMgr Class

The CSPIMgr implements software simulations of each *extended* XMCSPI function. The CSimpleDriver object is used to perform all *core* XMCSPI operations used in the simulations. The definition of the CSPIMgr class is as follows.

```

class CSPIMgr
(
public:
    //---- Constructors & Destructors ----
    CSPIMgr( void );
    ~CSPIMgr( void );

    //---- Initialization ----
    DWORD Initialize( LPCLSID lpDriverCLSID ); _

    //---- Extended SPI Methods ----
    DWORD FireExtSPI( XMC_EXT_SPI extspiID, ... );

private:
    //---- Private Ext SPI Simulation Functions ----
    :
    //---- Private Data ----
    CSimpleDriver* m_simpleDriver;
);

```

5.4.3 CSimpleDriver Class

The CSimpleDriver is used to communicate with the XMC Driver that performs the *core* XMCSPi functions. All OLE operations are encapsulated within this object. The definition of the CSimpleDriver is as follows.

```
class CSimpleDriver
{
public:
    //---- Constructors & Destructors ----

    CSimpleDriver( void );
    ~CSimpleDriver( void );

    //---- Initialization ----

    DWORD Initialize( LPCLSID lpDriverCLSID );
    DWORD LoadCoreInterfaces( void );
    DWORD LoadExtInterfaces( void );

    //---- Actions ----

    DWORD FireCoreSPI( XMC_CORE_SPI corespID, ... );
    DWORD FireExtSPI( XMC_EXT_SPI extspiID, ... );

private:
    //---- Private Data ----

    LPFNCORESPI m_rgCoreSPI[ XMC_CORE_SPI_COUNT ];
    BOOL m_bCoreSPIEnabled;

    LPFNEXTSPI m_rgExtSPI[ XMC_EXT_SPI_COUNT ];
    BOOL m_bExtSPIEnabled;
};
```